

Project Part 2

iae225

Part A

Following the procedure we studied, convert the ER diagram to a relational database schema diagram. Remember to include all PRIMARY KEY and FOREIGN KEY constraints that follow from the conversion rules of ER to schema.

Assume bold means pk

Answer

Category(**mainCategory**, **subCategory**, notes)

Item(**ItemID**, iDescription, photo, color, isNew?, hasPieces?, material, mainCategory, subCategory)

ItemCategory(**itemID**, **mainCategory**, **subCategory**)

ItemCategory(mainCategory) references Category(mainCategory)

ItemCategory(subCategory) references Category(subCategory)

ItemCategory(itemID) references Item(itemID)

Location(**roomNumber**, **shelfNumber**, shelfDescription)

Piece(**pieceNum**, **itemID**, pDescription, length, width, height, roomNumber, shelfNumber, storageNotes)

Piece(itemID) references Item(itemID)

Piece(roomNum) references Location(roomNumber)

Piece(shelfNumber) references Location(shelfNumber)

Person(**userName**, password, fname, lname, email)

PersonPhones(**userName**, **phoneNumber**)

PersonPhones(username) references Person(username)

DonatedBy(**donator**, **donation**, donationDate)

DonatedBy(donator) references Person(userName)

DonatedBy(donation) references Item(itemID)

Role(**roleID**, roleDescription)

Act(**userName**,**roleID**)
 Act(userName) references Person(userName)
 Act(roleID) references Role(roleID)

 Order(**orderID**, orderDate, orderNotes, supervisor, recipient)
 Order(supervisor) references Person(userName)
 Order(recipient) references Person(userName)

 Delivered(**deliveredBy**, **deliveredOrder**, orderStatus, orderDate)
 Delivered(deliveredBy) references Person(userName)
 Delivered(deliveredOrder) references Order(orderID)

 ItemIn(**orderItem**, **orderID**, found)
 ItemIn(orderItem) references Item(itemID)
 ItemIn(orderID) references Order(orderID)

Part B

```

CREATE SCHEMA IF NOT EXISTS Project2;

use Project2;

drop table if exists Category;
drop table if exists Item;
drop table if exists Location;
drop table if exists Piece;
drop table if exists Person;
drop table if exists PersonPhones;
drop table if exists DonatedBy;
drop table if exists Roles;
drop table if exists Act;
drop table if exists Orders;
drop table if exists Delivered;
drop table if exists ItemIn;

create table if not exists Category(
    mainCategory varchar(64),
    subCategory varchar(64),
    notes varchar(255),
    primary key (mainCategory, subCategory)
);

create table if not exists Item(
    itemID integer,
    itemDescription varchar(255),
    itemPhoto varbinary(8000),
    itemColor varchar(16),
    isNew boolean,
  
```

```

        hasPieces boolean,
        material varchar (128),
        primary key (itemID)
    );

create table if not exists ItemCategory(
    mainCategory varchar(64),
    subCategory varchar(64),
    itemID integer,
    primary key (mainCategory, subCategory, itemID),
    foreign key (mainCategory, subCategory) references Category(mainCategory, subCategory),
    foreign key (itemID) references Item(itemID)
);

create table if not exists Location(
    roomNumber integer,
    shelfNumber integer,
    shelfDescription varchar(255),
    primary key (roomNumber, shelfNumber)
);

create table if not exists Piece(
    pieceNum integer,
    itemID integer,
    pDescription varchar(255),
    plength float,
    pwidth float,
    pheight float,
    roomNumber integer,
    shelfNumber integer,
    storageNotes varchar(255),
    primary key (pieceNum, itemID),
    foreign key (itemID) references Item(itemID),
    foreign key (roomNumber, shelfNumber) references Location(roomNumber, shelfNumber)
);

create table if not exists Person(
    userName varchar(64),
    userPassword varchar(255),
    fName varchar(128),
    lName varchar(128),
    email varchar(255),
    primary key (userName)
);

create table if not exists PersonPhones(
    userName varchar(64),
    phoneNumber varchar(64),

```

```

        primary key (userName, phoneNumber),
        foreign key (userName) references Person(userName)
    );

create table if not exists DonatedBy(
    donator varchar(64),
    donation integer,
    donationDate date,
    primary key (donator, donation),
    foreign key (donator) references Person(userName),
    foreign key (donation) references Item(itemID)
);

create table if not exists Roles(
    roleID integer,
    roleDescription varchar(255),
    primary key (roleID)
);

create table if not exists Act(
    userName varchar(64),
    roleID integer,
    primary key (userName, roleID),
    foreign key (userName) references Person(userName),
    foreign key (roleID) references Roles(roleID)
);

create table if not exists Orders(
    orderID integer,
    orderDate date,
    orderNotes varchar(255),
    supervisor varchar(64),
    recipient varchar(64),
    primary key (orderID),
    foreign key (supervisor) references Person(userName),
    foreign key (recipient) references Person(userName)
);

create table if not exists Delivered(
    deliveredBy varchar(64),
    deliveredOrder integer,
    orderStatus varchar(16),
    orderDate date,
    primary key (deliveredBy, deliveredOrder),
    foreign key (deliveredBy) references Person(userName),
    foreign key (deliveredOrder) references Orders(orderID)
);

create table if not exists ItemIn(

```

```

orderItem integer,
orderID integer,
itemFound varchar(255),
primary key (orderItem, orderID),
foreign key (orderItem) references Item(itemID),
foreign key (orderID) references Orders(orderID)
);

```

take the additional insertions to be those made for the last question.

Part C

Question A

Record a new item that has been donated. It's a two-piece yellow sofa (category 'furniture', subcategory 'sofa'), donated by someone from your group. Their username should be their name as a single string, optionally with some numbers at the end. You may assign any itemID you'd like (or may look up how to use AutoIncrement). The pieces are the 'sofa body' and one 'cushion'. It will be stored in Room 5, without a shelf designated.

mainCategory	subCategory	notes
furniture	sofa	Sofas for Living ROoms
NULL	NULL	NULL

itemID	itemDescription	itemPhoto	itemColor	isNew	hasPieces	material
1	Two-piece yellow sofa	NULL	yellow	1	0	fabric
NULL	NULL	NULL	NULL	NULL	NULL	NULL

mainCategory	subCategory	itemID
furniture	sofa	1
NULL	NULL	NULL

pieceNum	itemID	pDescription	plength	pwidth	pheight	roomNumber	shelfNumber	storageNotes
1	1	sofa body	110	35	30	5	NULL	Stored in Room 5 without a designated shelf
2	1	cushion	33	15	5	5	NULL	Stored in Room 5 without a designated shelf
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

	userName	userPassword	fName	lName	email
	iae225	12345	ivan	aristy	iae225@stern.nyu.edu
	NULL	NULL	NULL	NULL	NULL

	donator	donation	donationDate
	iae225	1	2024-11-08
	NULL	NULL	NULL

```

insert ignore into Category values ('furniture', 'sofa', 'Sofas for Living Rooms');
insert ignore into Item values (1, 'Two-piece yellow sofa', null, 'yellow', true, false, 'fa
insert ignore into ItemCategory values ('furniture', 'sofa', 1);
insert ignore into Piece values (1, 1, 'sofa body', 110.0, 35.0, 30.0, 5, NULL, 'Stored in R
insert ignore into Piece values (2, 1, 'cushion', 33.00, 15.0, 5.0, 5, NULL, 'Stored in Room
insert ignore into Person values ('iae225', '12345', 'ivan', 'aristy', 'iae225@stern.nyu.edu
insert ignore into DonatedBy values ('iae225', 1, current_date);

```

Question B

Produce a list of all of the (pieces) of items in order #12345, along with their locations. This should have information that volunteers will find useful for locating the item when they're assembling the order, including the item IDs, their categories and subcategories, and the room and shelf where each piece is located. Optionally, you may include the description and other data.

Since we want to get items that might not have a designated piece we are doing outerjoins for the location.

```

insert ignore into Person values ('Someone', '321903210', 'some', 'person', 'anemail@nyu.edu
insert ignore into Orders values (12345, current_date, 'example', 'Someone', 'iae225');
insert ignore into ItemIn values (1, 12345, 'found what?');

```

```


select
    Piece.pieceNum as pieceNumber,
    Item.itemID as itemID,
    Item.itemDescription as itemDescription,
    Category.mainCategory as mainCategory,
    Category.subCategory as subCategory,

```

```

        Location.roomNumber as roomNumber,
        Location.shelfNumber as shelfNumber,
        Location.shelfDescription as shelfDescription,
        Piece.storageNotes as storageNotes,
        Piece.pDescription as pieceDescription,
        Piece.plength as length,
        Piece.pwidth as width,
        Piece.pheight as height
from
    Orders
join
    ItemIn on Orders.orderID = ItemIn.orderID
join
    Item on ItemIn.orderItem = Item.itemID
join
    Piece on Item.itemID = Piece.itemID
join
    ItemCategory on Item.itemID = ItemCategory.itemID
join
    Category on ItemCategory.mainCategory = Category.mainCategory and ItemCategory.subCategory = Category.subCategory
left join
    Location on Piece.roomNumber = Location.roomNumber and Piece.shelfNumber = Location.shelfNumber
where
    Orders.orderID = 12345
order by
    Item.itemID;

```

Result Grid													
Filter Rows: <input type="text" value="Search"/> Export: 													
pieceNumber	itemID	itemDescription	mainCategory	subCategory	roomNumber	shelfNumber	shelfDescription	storageNotes	pieceDescription	length	width	height	
1	1	Two-piece yellow sofa	furniture	sofa	5	110	35	30	Stored in Room 5 without a designated shelf	sofa body	110	35	30
2		Two-piece yellow sofa	furniture	sofa	5	33	15	5	Stored in Room 5 without a designated shelf	cushion	33	15	5